

学 号： WB2214161 专 业： 微电子科学与工程 姓名： 林灿

实验日期： 2023.3.31 教师签字： 成绩：

---

## 实 验 报 告 二

【实验名称】 数据及运算，选择结构程序设计

【实验目的】

- (1) 学会使用 C 的有关运算符，以及包含这些运算符的表达式的使用。
- (2) 了解 C 语言表示逻辑量的方法，学会正确使用逻辑运算符和逻辑表达式。
- (3) 熟悉 if~else 结构语句的使用。
- (4) 进一步熟悉 C 语言程序的编辑、编译、链接和运行的过程。

【实验内容】

1. 完成《上机指导 实验 2》实验步骤中的 (2) (3) (4)。
2. 完成《上机指导 实验 3》实验步骤中的 (1)。

【实验总结】

1. 记录实验遇到的问题和解决方法。
- 

### ● 实验 2

#### 步骤 (2)

代码编写如下：

```
#include<stdio.h>
void main(void)
{
    char c1='a',c2='b',c3='c',c4='\101',c5='\116';
    printf("a%c b%c\tc%c\tabc\n",c1,c2,c3);
    printf("\t\b%c%c",c4,c5);
}
```

运行程序，输出结果是



```
C:\Windows\system32\cmd.exe
aa bb cc abc
AN请按任意键继续. . .
```

第一个 printf 函数以字符形式输出了 c1,c2,c3 三个字符型变量

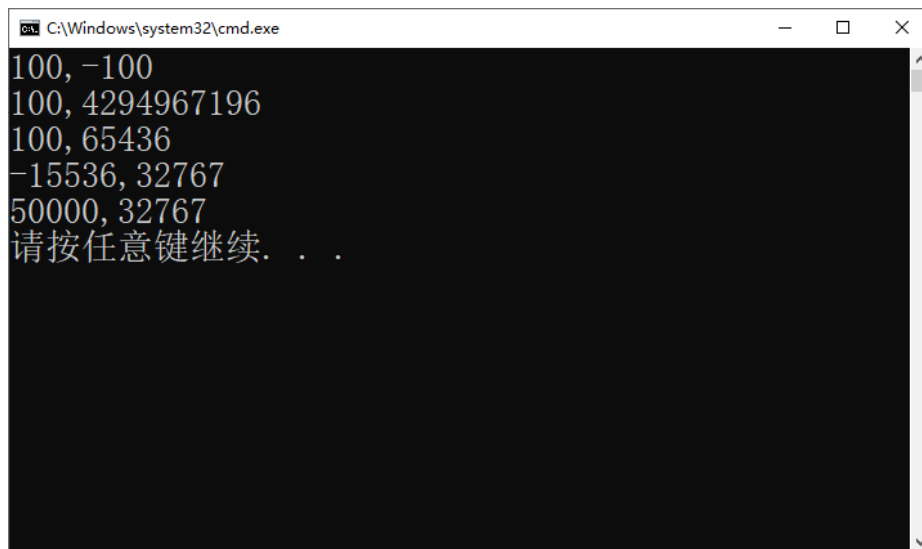
第二个 printf 函数输出了一个制表符，但随后的\b 转义符以退格形式删去了制表符

### 步骤（3）

代码编写如下：

```
#include<stdio.h>
void main(void)
{
    short a,b;
    unsigned short c,d;
    long e,f;
    a=100;
    b=-100;
    e=50000;
    f=32767;
    c=a;
    d=b;
    printf("%d,%d\n",a,b);
    printf("%u,%u\n",a,b);
    printf("%u,%u\n",c,d);
    c=a=e;
    d=b=f;
    printf("%d,%d\n",a,b);
    printf("%u,%u\n",c,d);
}
```

运行程序，输出结果是



```
C:\Windows\system32\cmd.exe
100,-100
100,4294967196
100,65436
-15536,32767
50000,32767
请按任意键继续. . .
```

- (1) 负整数-100 赋给无符号变量 d 后，会将这个负整数对应的补码的二进制形式存入这个无符号的变量中，-100 在计算机中以补码储存，其补码是 1111111110011100，对应整数 65436，故第三行输出的第二个数是 65436.
- (2) 将一个大于 32765 的长整数赋给短整型变量，会从低位向高位截取 16 位存储，且第十六为作为符号位，程序中将 32767 赋给了短整型变量 b，32767 对应的二进制是 0111111111111111，赋给 b 后 b 储存了 0111111111111111，符号位是 0，即是整数 32767 输出。
- (3) 与 (2) 的情况类似，同样会从低位向高位截取 16 位。

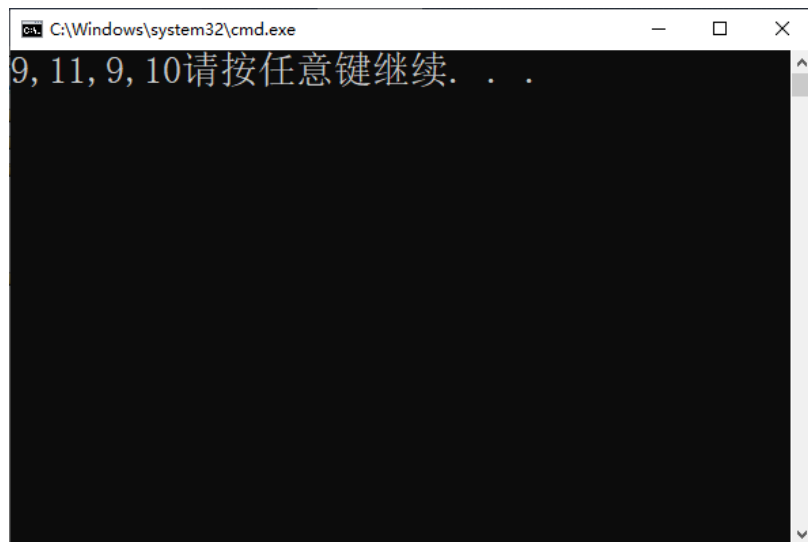
#### 步骤 (4)

代码编写如下：

```
#include<stdio.h>

void main(void)
{
    int i,j,m,n;
    i=8;
    j=10;
    m=++i;
    n=j++;
    printf("%d,%d,%d,%d",i,j,m,n);
}
```

运行程序，结果是



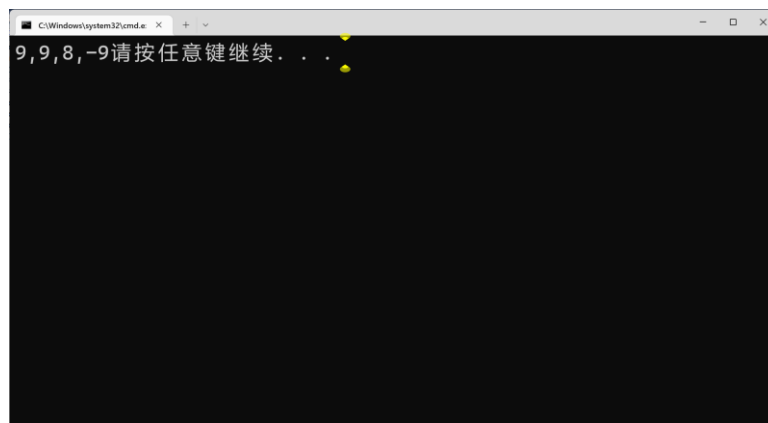
1) ++i 会先将 i+1 再赋给 m，而 j++是将 j 赋给 n 后再+1，所以 m 输出的结果是 9，而 n 输出的结果仍为 10。

2) 更改后，m, n 输出的结果分别是 8, 11, i 和 j 输出的结果不变。

3)

```
#include<stdio.h>
void main(void)
{
    int i,j,m=0,n=0;
    i=8;
    j=10;
    m+=i++;
    n--j;
    printf("%d,%d,%d,%d",i,j,m,n);
}
```

更改后，输出的结果是



### ● 实验 3

#### 步骤 (1)

代码编写如下：

```
#include<stdio.h>

void main(void)
{
    int a;
    scanf("%d",&a);
    if(a%2==0)
    {
        if (a%3==0)
            if(a%4==0)
                printf("1");
    }
    else
        printf("0");
}
```

运行程序，结果是



该程序的含义是，只有当输入的数字  $a$  除以 2, 3, 4 的余数均为零，才会输出 1，否则输出 0。当程序输出 1 时，实际上  $a$  能够同时被 2, 3, 4 整除，即  $a$  是 2, 3, 4 的公倍数，等价于  $a$  能够被 12 整除，所以 else 表示的情况是  $a$  不能被 12 整除。